



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/967,111	09/28/2001	Richard Francis Cormier	EMC01-11(01046)	8094

7590 10/04/2006

Barry W. Chapin, Esq.
CHAPIN & HUANG, L.L.C.
Westborough Office Park
1700 West Park Drive
Westborough, MA 01581

EXAMINER

ANYA, CHARLES E

ART UNIT PAPER NUMBER

2194

DATE MAILED: 10/04/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary

Application No.

09/967,111

Applicant(s)

CORMIER ET AL.

Examiner

Charles E. Anya

Art Unit

2194

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 13 July 2006.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-43 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-43 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
- Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
- Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
- ☐ Certified copies of the priority documents have been received.
 - ☐ Certified copies of the priority documents have been received in Application No. _____.
 - ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- 1) ☒ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☐ Information Disclosure Statement(s) (PTO/SB/08)
- Paper No(s)/Mail Date _____.

- 4) ☐ Interview Summary (PTO-413)
- Paper No(s)/Mail Date. _____.
- 5) ☐ Notice of Informal Patent Application
- 6) ☐ Other: _____.

WILLIAM THOMSON
SUPERVISORY PATENT EXAMINER

DETAILED ACTION

1. Claims 1-35 are pending in this application.

Claim Rejections - 35 USC § 103

2. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

3. **Claims 1,3,4,6-15,17, 20-30 and 36-42 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Pat. No. 6,560,606 B1 to Young in view of U.S. Pat. No. 6,842,856 B2 to Shenassa et al.**

4. As to claim 1, Young teaches in a computer system, a method for managing services associated with a plurality of plug-in modules, the method comprising the steps of: calculating a plug-in initiation order based upon the dependency list indicating respective plug-in services provided by, and required by, each plug-in module (“...determining...” Col. 8 Ln. 44 – 47, Col. 9 Ln. 18 – 24); and initiating service operation of plug-in modules according to the plug-in initiation order, such that if a first plug-in module provides a service required by a second plug-in module, the first plug-in module is initiated such that the service provided by the first plug-in module is available

to the second plug-in module when required by the second plug-in module (Col. 9 Ln. 24 – 33, Col. 14 Ln. 15 – 27).

Young is silent with reference to obtaining identities of a plurality of plug-in module and based on queries to the plurality of plug-in modules, retrieving a dependency list indicating respective plug-in services provided by, and required by, each plug-in module identified in the identities of a plurality of plug-in modules.

Shenassa teaches obtaining identities of a plurality of plug-in modules (“...identity information...” Col. 3 Ln. 55 – 62, “...identifying...” Col. 6 Ln. 29 – 32) and based on queries to the plurality of plug-in modules, retrieving a dependency list (array of plug-in descriptors) indicating respective plug-in services provided by, and required by, each plug-in module identified in the identities of a plurality of plug-in modules (“...getStartSequence()...” Col. 2 Ln. 56 – 61, “...retrieves...” Col. 6 Ln. 33 – 44).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of Shenassa and Young because the teaching of Shenassa would improve the system of Young by providing a process of starting plug-ins comprising assembling/arranging a startup sequence of a plurality of plug-ins (Shenassa Col. 1 Ln. 57 – 60, Col. 2 Ln. 58 - 61).

5. As to claim 3, Shenassa teaches the method of claim 1, wherein the step of retrieving a dependency list indicating respective plug-in services provided by, and required by, each plug-in module comprises the steps of: for each plug-in module identified in the identities of a plurality of plug-in modules, performing the steps of:

Art Unit: 2194

instantiating the plug-in module based upon a plug-in module definition associated with the identity of the plug-in module (“...lazy initiation...requested plug-in A...” Start Stage 160 Col. 3 Ln. 18 – 38, Col. 4 Ln. 25 – 44); receiving a dependency response from the plug-in module, the dependency response indicating respective plug-in services provided by, and required by, the plug-in module (“...plug-in context...” Col. 3 Ln. 56 – 67, Col. 4 Ln. 1 – 17); and storing identities of the respective plug-in services provided by, and required by, the plug-in module as identified in the dependency response in the dependency list (“...plug-in context...” Col. 3 Ln. 56 – 67).

6. As to claim 4, Young teaches the method of claim 3, wherein the step of: obtaining plug-in initiation information corresponding to the plug-in module definition associated with the identity of the plug-in module (“...read configuration data...” Col. 8 Ln. 41 – 47, Col. 14 Ln. 46 – 50).

Shenassa teaches instantiating the plug-in module comprises the steps of: instantiating the plug-in module based upon a plug-in module definition associated with the identity of the plug-in module (“...lazy initiation...requested plug-in A...” Start Stage 160 Col. 3 Ln. 18 – 38, Col. 4 Ln. 25 – 44) and passing the plug-in initiation information to the plug-in module for use by the plug-in module (“...getStartSequence...” Col. 5 Ln. 51 – 62).

7. As to claim 6, Young teaches the method of claim 1, wherein the step of calculating a plug-in initiation order based upon the dependency list comprises the step

of: arranging a placement of each plug-in module identified in the dependency list within the plug-in initiation order such that plug-in modules not requiring services provided by other plug-in modules are placed earlier in the initiation order and such that plug-in modules requiring services provided by other plug-in modules are placed later in the initiation order (Col. 15 Ln. 41 – 50).

8. As to claim 7, Young teaches the method of claim 6, wherein the step of arranging placement of each plug-in module identified in the dependency list within the plug-in initiation order comprises the steps of: analyzing the dependency list indicating respective plug-in services provided by, and required by, each plug-in module to determine which plug-ins provide services relied upon by other plug-in modules (Col. 8 Ln. 44 – 47); and creating, as the plug-in initiation order, at least one plug-in module dependency tree based on the step of analyzing, the at least one plug-in module dependency tree containing a hierarchical arrangement of nodes associated with respective plug-in modules, the hierarchical arrangement indicating the plug-in initiation order of the plug-in modules respectively associated with the nodes in the dependency tree (Col. 8 Ln. 44 – 47, Col. 9 Ln. 18 – 33, Col. 13 Ln. 41 - 64).

9. As to claim 8, Young teaches the method of claim 7, wherein the step of initiating service operation of plug-in modules according to the plug-in initiation order comprises the steps of: traversing the at least one plug-in module dependency tree according to the hierarchical arrangement of nodes and for each node encountered during the step

Art Unit: 2194

of traversing, initiating service operation of the respective plug-in module associated with that node (“...dependency graph...” Col. 8 Ln. 38 – 47, Col. 9 Ln. 18 – 33, Col. 13 Ln. 41 – 64).

10. As to claim 9, Young teaches the method of claim 8, wherein the step of initiating service operation of the respective plug-in module associated with that node comprises the steps of: forwarding, via a dependency available interface associated with the respective plug-in module, a list of initiated plug-in services of other plug-in modules that are currently available for use by the respective plug-in module (Col. 8 Ln. 41 – 43, Col. 14 Ln. 46 – 49).

11. As to claim 10, Shenassa teaches the method of claim 1, wherein the step of initiating service operation of plug-in modules according to the plug-in initiation order comprises performing, for each respective plug-in module in the plug-in initiation order, the steps of: determining, from a published list of services available by initiated plug-in modules, an identity of each initiated plug-in service required by the respective plug-in module (“...determines...” Col. 5 Ln. 51 – 57); forwarding to the respective plug-in module, via a dependency available interface associated with the respective plug-in module, the identity of each initiated plug-in service required by the respective plug-in module; receiving a list of services initiated by the respective plug-in module (“...getStartSequence...” Col. 5 Ln. 51 – 62); and adding the list of services provided by

Art Unit: 2194

the respective plug-in module to the published list of services (“...stored...” Col. 5 Ln. 63 – 67).

12. As to claim 11, Young teaches the method of claim 1, wherein the step of initiating service operation of plug-in modules according to the plug-in initiation order operates such that if the second plug-in module requires a service provided by the first plug-in module, the second plug-in module is initiated such that the service provided by the first plug-in module is available to the second plug-in module when required (Col. 9 Ln. 24 – 33).

13. As to claim 12, Young teaches the method of claim 1, wherein the first plug-in module is initiated via the step of initiating operation of plug-in modules prior to initiation of the second plug-in module (Col. 9 Ln. 24 – 33).

14. As to claim 13, Shenassa teaches the method of claim 1, wherein the first plug-in module is initiated via the step of initiating operation of plug-in modules after initiation of the second plug-in module, and wherein the second plug-in module includes a wait-state operation causing the second plug-in module to wait to provide the service offered by the second plug-in module until initiation of the first plug-in module (“...wait...” Col. 3 Ln. 32 – 38).

Art Unit: 2194

15. As to claim 14, Young teaches the method of claim 1, wherein the steps of obtaining, retrieving, calculating and initiating are performed by a multi-threaded plug-in manager and wherein the step of calculating a plug-in initiation order is performed by collectively operating a respective thread for each plug-in, each thread performing the step of initiating service operation of at least one plug-in module when all services required by that plug-in module are available (Col. 13 Ln. 33 – 36, Col. 15 Ln. 58 – 62).

16. As to claims 15,17 and 20-28, see the rejection of claims 1,3 and 6-14 respectively.

17. As to claim 29, see the rejection of claim 1 above.

18. As to claims 30, see the rejection of claims 1 and 13 above.

19. As to claim 36, Young teaches the computer program product as in claim 29, wherein initiating service operation of the plug-in modules includes: initiating at least partial concurrent execution of the first plug-in module and the second plug-in even though the second plug-in module requires the service provided by the first plug-in module (“...any order...” Col. 9 Ln. 45 – 50, “...forking plug-ins...redirect...” Col. 14 Ln. 1 – 6).

Art Unit: 2194

20. As to claim 37, Shenassa teaches the computer program product as in claim 29, wherein initiating service operation of the plug-in modules includes initiating execution of a first plug-in module that implements a wait state operation, the wait state operation causing the first plug-in module to signal to a dependent second plug-in module that a respective service is not available (“...not immediately available...wait...” Col. 3 Ln. 32 – 38).

21. As to claim 38, Young teaches the method as in claim 1, wherein initiating service operation of the plug-in modules includes initiating at least partial concurrent execution of the first plug-in module and the second plug-in module (“...simultaneously...parallel...” Col. 50 – 60).

22. As to claim 39, Young teaches the method as in claim 38, wherein the second plug-in module depends on a given service provided by the first plug-in module (“...any order...” Col. 9 Ln. 45 – 50, Col. 13 Ln. 51 – 58).

23. As to claim 40, Shenassa teaches the method as in claim 39, wherein initiating service operation of the plug-in modules enables communication between the first plug-in module and the second plug-in module to enable each other to indicate when the given service is available (“...lazy initiation...available...” Col. 3 Ln. 32 – 38).

Art Unit: 2194

24. As to claim 41, Shenassa teaches the method as in claim 40, wherein during the at least partial concurrent execution, the first plug-in module initiates a wait state and notifies the second plug-in module that the first module is for service processing (“...notified...” Col. 4 Ln. 1 – 17).

25. As to claim 42, Shenassa teaches the method as in claim 40, wherein the at least partial concurrent execution, the first plug-in module initiates a wait state and notifies the second plug-in module that the first plug-in module is waiting for the second plug-in module to provide a particular service (Col. 3 Ln 32 – 38, Col. 4 Ln. 1 – 18).

26. Claims 2,5,16,18,19,31-35 and 43 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Pat. No. 6,560,606 B1 to Young in view of U.S. Pat. No. 6,842,856 B2 to Shenassa et al. as applied to claims 1 or 3 or 15 or 29 above, and further in view of U.S. Pat. No. 6,704,750 B1 to Asazu.

27. As to claim 2, Shenassa and Young are silent with reference to the method of claim 1, wherein the step of obtaining identities of a plurality of plug-in modules includes the steps of: receiving a list of services to be started within the computerized device/determining, for each service in the list of services, a respective plug-in module definition that can provide that service; and placing the identity of each plug-in module definition determined in the step of determining into the identities of the plurality of plug-in modules.

Asazu teaches the method of claim 1, wherein the step of obtaining identities of a plurality of plug-in modules includes the steps of: receiving a list of services to be started within the computerized device/determining, for each service in the list of services, a respective plug-in module definition that can provide that service; and placing the identity of each plug-in module definition determined in the step of determining into the identities of the plurality of plug-in modules (Col. 12 Ln. 12 – 23).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of Asazu, Shenassa and Young because the teaching of Asazu would improve the system of Shenassa and Young by providing a means for efficient retrieval of plug-ins (Asazu Col. 12 Ln. 17 – 23).

28. As to claim 5, Asazu teaches the method of claim 3, wherein the step of instantiating the plug-in module comprises the step of: querying a dependency interface associated with the plug-in module with a dependency query to obtain the dependency response from the plug-in module (Col. 12 Ln. 7 – 23).

29. As to claim 16, see the rejection of claim 2 above.

30. As to claims 18 and 19, see the rejection of claims 4 and 5 respectively.

31. As to claim 31, Asazu teaches the computer program product as in claims 29, wherein the processor further performs operations of: determining a list of plug-in

Art Unit: 2194

service by a software application; and querying a set of plug-in modules to identify services provided by the set of plug-in modules (Col. 11 Ln. 60 – 65, Col. 12 Ln. 17 – 23).

32. As to claim 32, Shenassa teaches the computer program product as in claims 31, wherein the processor further performs operations of: in response to querying the set of plug-in modules, identifying plug-in modules not identified by the software application as being necessary to carry out execution of an operation on behalf of the software application (“...flag...set...true...” Col. 7 Ln. 1 – 4).

33. As to claim 33, Shenassa teaches the computer program product as in claim 32, wherein the processor further performs operations of: initiating service operation of plug-in modules on the processor according to an order other than the plug-in initiation order, such that if a third plug-in module provides a service required by the fourth plug-in the third plug-in module being initiated after initiation of the fourth plug-in module, the third plug-in module initiating a wait state operation causing the third plug-in to wait to provide the service offered by the third plug-module until instantiation of the fourth plug-in module (“...wait...” Col. 3 Ln. 32 – 38).

34. As to claim 34, Shenassa teaches the computer program product as in claim 32, wherein the processor initiates execution of the first plug-in module before execution of the second plug-in module, the first plug-in module initiating a wait state operation

Art Unit: 2194

resulting in signaling to the second plug-in module, the signaling indicating that a respective service of the first plug-in module is not yet available to the second plug-in module (“...not immediately available...wait...” Col. 3 Ln. 32 – 38).

35. As to claim 35, Asazu teaches the computer program product as in claims 34, wherein the processor further performs operations of: maintaining a list of services for a set of plug-in modules currently able to provide respective services; and publishing the list of services for the software application to identify instantiated plug-ins currently providing the respective services (Col. 12 Ln. 12 – 23).

36. As to claim 43, Asazu teaches the method as in claim 1, further comprising: utilizing services of initiating plug-in modules for purposes of managing resources associated with a respective storage area network (figure 1 Col. 5 Ln. 18 – 67).

Response to Arguments

Applicant's arguments with respect to claims 1,3,10,13,31,32-34, xxx have been considered but are moot in view of the new ground(s) of rejection.

Applicant's arguments filed 7/13/06 have been fully considered but they are not persuasive.

Applicant argues in substance that the Young prior art does not teach or suggest claim limitation of claim 9 because its configuration information does not identify a list of initiated plug-in services of other plug-in modules that are currently available for use.

Art Unit: 2194

Applicant also alleges that the flow of information in the disclosure of the Young prior art is opposite that of the claimed invention.

Examiner respectfully traverses Applicant's arguments:

The forwarding step of claim 9 does not identify the direction of flow or to/from where the forwarding originated/destined and therefore erroneous to argue the flow of information in the disclosure of the Young prior art is opposite that of the claimed invention.

The configuration information of the Young prior art does indeed identify a list of initiated plug-in services of other plug-in modules that are currently available for use. The configuration information holds associated with the dependency graph and specifies the number of other plug-ins on which each plug-in depends. The configuration information is used to execute the plug-ins in a correct order (Column 14 lines 46 – 49).

Conclusion

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, William Thomson can be reached on (571) 272-3718. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Charles E Anya
Examiner
Art Unit 2194

cea.


WILLIAM THOMSON
SUPERVISORY PATENT EXAMINER